Anomaly Detection in Orthogonal Metal Cutting based on Autoencoder Method

Dawoua Kaoutoing Maxime
Laboratoire Génie de Production
Ecole Nationale d'Ingénieurs
Tarbes, France
mdawouak@enit.fr

Houé Ngouna Raymond
Laboratoire Génie de Production
Ecole Nationale d'Ingénieurs
Tarbes, France
raymond.houe-ngouna@enit.fr

Pantalé Olivier
Laboratoire Génie de Production
Ecole Nationale d'Ingénieurs
Tarbes, France
olivier.pantale@enit.fr

Beda Tibi
Faculté de Sciences - PAI
Université de Ngaoundéré
Ngaoundéré, Cameron
tbeda@yahoo.fr

***Abstract*—The choice of appropriate cutting conditions is widely acknowledged as a key performance indicator for efficient machining, since it allows to mitigate tools wear. In precision machinery, tool wear can indeed lead to poor surface quality and even affect the dimensions of the final product. In such a context, the cutting and feed forces, along with other parameters are affected and are not optimal. It is therefore necessary to detect any anomaly and provide the operators in the plant with a decision support, allowing to monitor the machining conditions in order to predict the outputs values and anticipate the wear's damage on the whole cutting process. In this article, cutting speed, cutting angle, cutting width and feed depth are the input parameters. Using the extended-Oxley analytical model of orthogonal metal cutting, a sample of cutting conditions has been simulated in order to analyze the optimality of the cutting force, the feed force and the internal temperature, based on the autoencoder method. Given a threshold for decision, the results allow to identify abnormal parameters and provide significant insights for operators, allowing them to avoid error and make the best choices of the inputs for optimal cutting conditions.**

Keywords—Orthogonal metal cutting, tool conditions monitoring, anomaly detection, simulation, autoencoder.

I. INTRODUCTION

The present work is based on anomaly detection in orthogonal metal cutting. In such a context, the cutting tool has a straight cutting edge, which is perpendicular to the cutting velocity direction (see Fig. 1). The simplicity of such context allows researchers to describe mechanics of machining process in order to understand the chip formation. Besides, it provides some advantages which are of great interest for producing a reasonably good approximation of material responses to cutting operation, under many conditions. As depicted in Fig. 1, V is the cutting speed, α the cutting angle, w the cutting width, and t_1 the feed depth. These four quantities will be referred further as the cutting parameters, considered in this work as the inputs of the cutting process and also the inputs of the machine learning model provided. Oxley investigated the influence of the material properties and the effect of strain, strain rate and temperature on the chip formation process [1]. The achievement of the author and co-workers' studies is crystallized under the *Oxley predictive machining theory,* which provides a class of theoretical relationships between orthogonal machining process variables and workpiece material properties, tool geometry and cutting conditions.

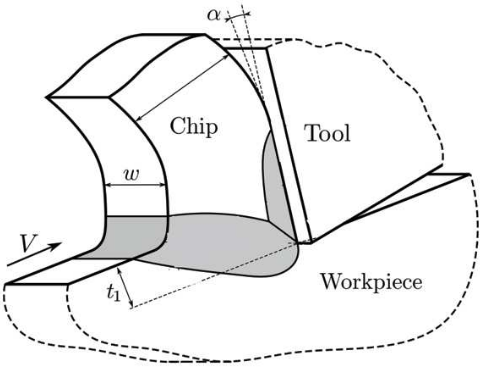

Fig. 1. Orthogonal metal cutting configuration.

From the Oxley predictive model, machining characteristics such as temperature in deformation zones, cutting forces, chip geometries, etc. can then be obtained without any need to experiment the cutting process, based on the inputs data of workpiece material properties, tool geometry, cutting conditions. In orthogonal metal cutting, the optimal choice of the input parameters is widely acknowledged as a key success for efficient machining. In the literature, several studies have been dedicated to tool conditions machining [2][3]. Monitoring these conditions is necessary to anticipate any fault and may help machining companies to provide high-quality product. As mentioned in [4], a tool monitoring system aims at fulfilling the following purposes: *(i)* fault detection for cutting and machine tool, *(ii)* check and safeguard machining process stability, *(iii)* machining tolerance maintenance on the workpiece to acceptable threshold, and *(iv)* machine tool damage prevention. The problem is that there are several interrelated interactions between machines, tools, workpieces, material, measurement systems, humans, the environment of the cutting process, etc., which make it complex; therefore, it become hard to analyze the behavior of the process in order to provide significant insights to decision makers. Thanks to the democratization of sensors, these latter are increasingly developed and employed to ensure efficient machining and protect workers [2]. As a consequence, an intelligent monitoring system is being required in order to provide efficient answers to the purposes mentioned above.

Due to growing demand for high-quality product, machining companies all around the world are increasingly facing internal issues regarding their cutting processes, which recommend the development of reliable monitoring systems. One key issue is indeed concerned with tool wear, which can affect the product quality and the security as well. In many cases, as mentioned in [3], the cost of the workpiece is higher than the price companies would pay to change a tool, which justifies the need for a monitoring system allowing to anticipate. One way, within the whole monitoring process, is to detect anomalies as earlier as possible.

In contrast to an *outlier*, which is a legitimate observation (far from the average values), an *anomaly* represents an illegitimate data point generated by a process different from the one which generated the rest of the data. Both are significantly different from the remaining data. As mentioned in [5], Hawkins [6] formally defined anomaly as *"an observation which deviates so much from the other observations as to arouse suspicions that it was generated by a different mechanism"*. So, analyzing and detecting anomalies may be considered as a promising direction for identifying suspicious behavior of the machining process, and therefore, a strategy to anticipate any failure, due for instance to tool wear. To address this issue, in the context of metal cutting, most studies of the literature, driven by the use of sensors, have provided good results for performing robust monitoring systems [2]. According to the authors in [7], an accurate tool conditions monitoring could even increase savings up to 40%. However, due to the complexity of the cutting context (resulting from the interrelationships between many factors, as noted above), sensors themselves may not be able to perform complex analyses that go beyond the comparison between measured values and admissible thresholds. Although intelligent sensors for complex analyses are being developed, their use is not yet well established.

Data-driven methods are then needed to analyze data, including those acquired through sensing devices. In the context of metal cutting, such analyses are generally based on real-life experiences, with few data acquired, certainly due to the cost of the experiments. For that reason, in order to take advantage of the learning methods (which can analyze massive data), and using the Oxley predictive model which allows to predict the machining features without any need to experiment the process (as mentioned above), the cutting parameters of this work have been simulated in order to provide enough information to train the proposed machine learning model. Within the frame of metal cutting, the present study is particularly concerned with industrial damage detection, which refers to different faults and failures in complex industrial systems, such as abnormal consumption, abnormal temperature.

The rest of the article is organized as follows: in section II, a brief summary of the work related to this study is presented, including tool wear monitoring, and anomaly detection methods. Section III provides the proposed methodology for anomaly detection in the context of orthogonal metal cutting, followed by its application, which is described in section IV; the main choices and methods used for that purpose will then be explained, along with useful discussions based on the results provided. Finally, the last section concludes the work and provides some directions of improvements.

II. Related Work

As stated above, the present work is concerned with the proposal of a machine learning method allowing to detect anomalies in metal cutting process. The aim is to prevent tool wear through monitoring system, consisting in extracting useful knowledge from the cutting input data. A brief summary of the related work is presented in the following.

A. *Tool wear monitoring*

In [8], tool wear monitoring has been classified into two categories of methods: direct and indirect methods. With the first ones, tool wear is directly measured, for instance using computer vision, laser beams, electrical resistance, etc. Due to their lack of technical and economic efficiency, the author noted that such methods are not attractive to industrials. For that purpose, they are mostly used in laboratory instead [2]. Some practical limitations noted are, for instance, access problems during machining, illumination, use of cutting fluid and, in a broad sense, the fact that external device introduced within a physical system may disturb the normal functioning of the concerned system; as a consequence, such intrusion can corrupt the expected measurements. With indirect methods, auxiliary quantities are measured: the actual quantity is then deduced through correlations' assessment. In contrast to the previous category of methods, this latter is recognized in the literature as less complex for assessing tool wear and even allows continuous monitoring via sensing devices. In that purpose, different sensor technologies are used, depending on the machining characteristic to assess [2]: (1) motor power and current measurement technology, which includes the relationship between input current/power and output force/torque; (2) force and torque measurement technology, based on sensing elements that convert the applied force into deformation of an elastic element; (3) acoustic emission measuring technology, based on piezoelectric sensor technology; (4) vibration using piezoelectric transduction technology; (5) cutting temperature resulting around the cutting tool edges, which influences the rate and mode of cutting tool wear, the friction between chip and the cutting tool, etc.

In this study, the main aim is not to measure the wear itself, but to provide and validate a methodology for analyzing machining anomalies, by monitoring the cutting conditions that may generated tool wear (for instance, the conditions that produce high internal temperature). Thanks to the Oxley predictive model (as mentioned earlier), tool wear is deduced through abnormal values of the machining features that are predicted, given some cutting conditions. The main issue is then to find an appropriate method for detecting an anomaly.

B. Anomaly detection methods

Anomaly detection is a well-known approach applied in several real-world applications, including credit card fraud, medical diagnosis, marketing, network intrusion, industrial damage detection, etc. As mentioned in [9], an anomaly might result from various reasons and its detection is related to, but distinct from removing unwanted noise from data. This latter is defined as a phenomenon in data that is not of interest to the analyst. Therefore, denoising is driven by the need to eliminate the unwanted objects before any analysis, while anomaly represents a pattern that does not conform to normal behavior.

In [9], the authors have identified several challenging factors regarding anomaly detection, among which: defining the region of normal behavior; how to treat observation that lies close to the boundary; how to identify an anomaly taking into account the evolution of the normal behavior which might change in the future; the availability of labeled data allowing to distinguish among normality and abnormality (i.e. the use of supervised machine learning methods); the variability of the notion of anomaly which differs from one application domain to another (e.g. small deviation in body temperature might be abnormal, while similar fluctuations in the stock market might be considered as normal); how to efficiently remove noise in data that tend to be similar to the actual anomalies, etc. Based on these challenges, detecting anomalies cannot be performed through a single general model. Specific formulation of the context has to be defined, including the nature of input data, the availability of labeled data, the type of anomalies to be detected, the output data (scores or labels) through which anomalies are reported, etc.

The main anomaly detection methods are summarized in the following, as reviewed in a survey presented in [9].

1) Classification-based

Classification in learning methods is based on a set of labeled data that train a model (the classifier); then, using a test set (composed of unlabeled data), it tries to classify them into one of the classes defined in the training set. Anomaly detection methods based on classification operate in a similar process: a training phase learns a classifier with the labeled data (normal and abnormal), followed by a test phase which classifies a test set data into normal and abnormal, using the learned classifier. As noted in the survey, the process works under the assumption that a classifier that is able to distinguish among normal and abnormal classes can be learned in the given feature space.

A variety of classification methods used in the context of anomaly detection has been identified by the authors of the survey, among which: *neural networks* methods as in [10][11] for one-class anomaly detection, *Bayesian networks* as in [12] for disease outbreak detection, in [13] for network intrusion, and in [14] for novelty detection in video surveillance, *support vector machine* (SVM) as applied in [15] for anomaly detection in audio signal data, in [16] for novelty detection in power generation plants, etc.

The main advantage of these methods is their ability to distinguish between data belonging to different classes; but an issue (its main drawback) is the availability of labeled data, which is not always the case in practice.

2) Nearest neighbor-based

A nearest neighbor analysis is used in anomaly detection under the assumption that normal data occur in dense neighborhood, while abnormal ones are far from their closest neighbors. Based on this assumption, there is a need to define distance or similarity measure between data. The only two properties required for such measure is that it should be positive-definite and symmetric.

Methods using nearest neighborhood can be classified into two different categories: (1) methods that use the distance for a data to its *k-th* nearest neighbor as the anomaly score, (2) those that compute the relative density of each data to define its anomaly score (i.e. low density reflects an anomaly). The first category has for instance been applied in [17] to detect land mines from satellite ground images, while the second has been proposed in [18] to detect sequential anomalies in protein sequences or in [19] for detecting spatial anomalies in climate dataset.

The main advantage of the nearest neighbor methods is that they do not need labeled data to perform anomaly detection; an issue is then to choose the appropriate value *k* for determining the neighborhood. A known drawback is that, if the dataset at hand contains normal instances with few close neighbors, the method can fail to identify anomaly properly, resulting in keeping abnormal data into normal class. The same problem occurs if the dataset has abnormal instances that have close neighbors in a normal class.

3) Clustering-based

A clustering method is an unsupervised learning method that can be applied in the context of anomaly detection. Three categories can be distinguished in the literature.

The first one assumes that normal data belong to a cluster, while anomalies do not. Methods based on this assumption directly apply a clustering algorithm. Their main drawback is that, since clustering main focus is to find clusters (not anomalies), they are not optimized for anomaly detection.

The second one assumes that normal data lie close to their closest centroid, while anomalies are far away. Two main steps are used: firstly, a clustering algorithm is performed, and secondly, the distance of each data to its closest centroid is computed and defined as its anomaly score. Several types of clustering algorithms have been applied in the literature, based on these two steps, among which: self-organizing map (SOM) implemented in a semi-supervised mode in [20] for intrusion detection applications, or in [21] for fraud detection. A known drawback of these methods is that the algorithms fail to detect anomalies that form a cluster, which is solved by the third category of methods described next.

The third category assumes that normal data belong to large and dense clusters, while anomalies either belong to small or sparse clusters. The associated methods use the size and/or density of the clusters as anomaly score, which is then compared to a pre-defined threshold, as in [22].

4) Other methods

Many other methods have been proposed in the literature for anomaly detection, including statistical methods (parametric and non-parametric), information theory-based, spectral-based, etc.

In short, it appears that each method has its own advantages and drawbacks, and that the appropriate choice depends on the context of the study: for the present work, regarding the input data to use for anomaly detection, a simulation-based approach has been adopted; the aim is to simulate a realistic machining process, using the Oxley analytical predictive model. This allows to generate massive data in order to test the method used, which is not the case is most real-world applications in metal cutting, generally based on few data acquired through costly experiments. For the present study, an unsupervised method has been adopted (since the simulated data are not labeled). In contrast to most of the literature studies, a neural network model has been implemented, due its ability to cope with complex relationships between inputs, and using data that have been simulated according to a process described further. The theoretical background of this proposal is explained in the following.

III. Theoretical Background of the Proposed Learning Method for Anomaly Detection

The proposed method for anomaly detection in metal cutting is based on an autoencoder (AE) model, which is neural network-based. The theoretical background of the concerned model, along with the main choices adopted to define the model are summarized in the following.

A. Overview of autoencoder model

An AE is a kind of neural network that attempts to reconstruct the input at the output (as illustrated in Fig. 2). It is not actually a typical method for anomaly detection, commonly used in deep learning applications for data denoising (namely image denoising), or for dimensionality reduction in data visualization. In such cases, the dimension of inputs is high (hundreds or more), and the method tries to synthetize them into fewer parameters (in the hidden layer). However, the machining process, as considered in the present study, is based on only the four following parameters: cutting speed *(V),* cutting angle *(α),* cutting width *(w),* and feed depth *(t_1)*. Besides, the concerned process is simulated (using the Oxley predictive model), which might then differ from a real-life machining context: many other parameters such as the environmental temperature of the plant might indeed has an influence to the process. Based on that, an AE with more nodes in the hidden layers (than in the input and output) is required, in such a way that more tool conditions than the four parameters used in the input are implicitly considered in the model. Therefore, the simulation approach adopted can provide a realistic behavior to the frame of the considered simulated machining process.

The theoretical background of an AE model can be defined as follows (assuming one hidden layer, for simplicity):

Let $x \in \mathbb{R}^n$ be an input parameter. The hidden representation $h(x) \in \mathbb{R}^m$ (with $m > n$ in this study) is defined as in (1),

$$h(x) = f_1(W_1 x + b_1) \tag{1}$$

where $W_1 \in \mathbb{R}^{nxm}$ is a *weight* matrix, and $b_1 \in \mathbb{R}^m$ a *bias* vector, both associated to the input, and f_1 a non-linear and differentiable *activation function*. The non-linearity property makes the model learn and perform complex relationships between the input data. Some examples of activation functions are the *sigmoid function (σ)* depicted in (2), the *hyperbolic tangent* (*tanh*) function depicted in (3):

$$\sigma(x) = \frac{1}{1+e^{-x}} \tag{2}$$

$$tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \tag{3}$$

The latent variable h is mapped back into a reconstruction vector $\hat{x} \in \mathbb{R}^n$ in the output layer using the formula in (4):

$$\hat{x} = f_2(W_2 h(x) + b_2) \tag{4}$$

where $W_2 \in \mathbb{R}^{mxn}$ and $b_2 \in \mathbb{R}^n$ have the same meanings as in (1) regarding the output layer, and f_2 is an output function (for instance, the sigmoid function for classification problems, or the linear identity function for regression problems).

B. The learning process

As in any neural network model, learning will occur when the weights parameters are adapted to minimize a function $L(x, f_2(f_1(x)))$. L is a loss function penalizing $f_2(f_1(x))$ for being different from x. An example, commonly used in regression problem as in the context of the present study, is the mean squared error, which represents an average reconstruction error. The optimized parameter $\theta = (W_1, W_2, b_1, b_2)$ is obtained after minimizing the loss function. In the specific context of anomaly detection, a model trained on normal data fails to reproduce anomalies in the dataset, providing a high reconstruction error. The idea is then to use the reconstruction error as a score indicator for detecting anomalies. An issue is therefore to determine a threshold for the anomaly test, besides the choice of the hyper-parameters of the model architecture, which is explained further regarding the present study.

A basic pseudo-code of a learning process, using stochastic gradient descent (the most commonly adopted) is given below, based on one of the LeCun and co-workers' achievements [23]:

```
1.  Initialize the weights (Wj) and the biases (bj)
2.  Iterate until convergence criterion is reached:
        a.  Get the training sample
        b.  Update all weights and biases as in (5)
            and (6):
```

$$w_{jk} := w_{jk} - \alpha \frac{\partial L(W, B)}{\partial w_{jk}} \tag{5}$$

$$b_{jk} := b_{jk} - \alpha \frac{\partial L(W, B)}{\partial b_{jk}} \tag{6}$$

The constant α is the *learning rate*, which controls how much the algorithm adjust the weights (i.e. the lower the value, the slower the algorithm travels along the loss gradient).

A well-known problem, when applying autoencoder is that, it can fail to learn anything if the encoder f_1 and the decoder f_2 are given too much capacity [24]: in such a case, the model may just keep reproducing the same input values at the output. To avoid this, several penalty functions can be added to the loss function, which are called "regularization" functions, among which the *sparsity penalty* $\Omega(h)$. As noted in [24], training a model with sparsity penalty allows to perform a model that has learned useful features as *"a byproduct"*.

An issue here is to define the appropriate sparse penalty in order for the model to perform well. A loss function with sparse penalty looks like (7):

$$L(\theta) = L\left(x, f_2\big(f_1(x)\big)\right) + \Omega(h) \quad (7)$$

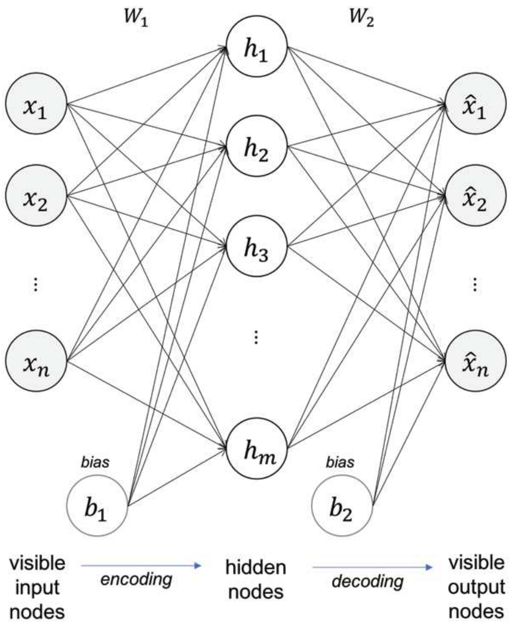

Fig. 2. An Autoencoder architecture.

Another kind of regularization, aiming at avoiding overfitting (and which improves generalization), is the lasso regularization l_1 introduced to the loss function as in (8):

$$L(\theta) = L\left(x, f_2\big(f_1(x)\big)\right) + \lambda_1 R_1(\theta) \quad (8)$$

where $R_1(\theta) = \|\theta\|_1 = \sum_j |\theta_j|$ (i.e. the sum of all of the $\|.\|_1$ norm of the weights and biases in the network). Similar regularization can be used, based on the square of the $\|.\|_2$ norm (i.e. Euclidean norm) of the weights and biases, called *ridge* regularization. In practice, the constant λ_1 in lasso regularization is chosen as small as possible (for instance *1e-5*). An issue is to make the best choice in order for the model to perform well. Depending on the architecture of the neural network, several penalty functions can be added in conjunction to the loss function, but this may slower the computations. In summary, an interesting property of penalty functions is that, during the training process, they encourage the model to have other features besides their capacity to copy the input to the output.

IV. Application of the Autoencoder Method for Anomaly Detection in Metal Cutting

Metal cutting in the present study is based on the machining context illustrated in Fig. 3: as noted above, 4 cutting parameters are considered (the cutting speed, the cutting angle, the cutting width and the feed depth). These are the parameters associated to the input of the machining process. Thanks to the extended-Oxley predictive model, several machining features can be predicted without any experiment, among which the cutting force, the feed force, the internal temperature (considered as the output of the machining process in the present work).

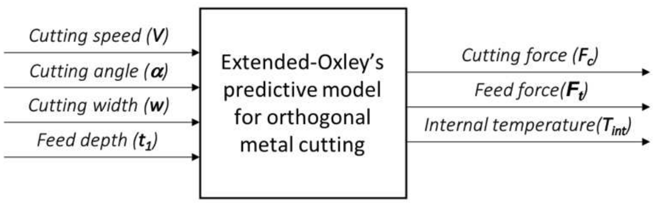

Fig. 3. Machining context of the study.

The predictive model considered is an analytical model based on an equilibrium system, consisting in a continuous optimization problem; more precisely, it is a minimization of the cutting force (of the machining process), under two conditions: (1) equilibrium of normal stress, and (2) equilibrium of shear stress. An optimization algorithm to solve this problem has been proposed in another study in which its mathematical formulation has also been deeply described.

In contrast to several works in the literature, and as explained earlier, tool wear is not directly measured in the context of this study. It is deduced from abnormal machining features (i.e. the output elements of the machining process, as those depicted in Fig. 3). In short, the machining process considered is simulated, based on the methodology presented in the following.

A. The proposed methodology for anomaly detection

The methodology proposed for detecting anomalies in the present work is illustrated in Fig. 4, based on four main phases described in the following.

The first phase is concerned with ***data simulation***. As noted above, in the context of this study, tool wear is not directly measured (for instance, using sensing devices as many studies of the literature), but deduced from abnormal machining features such as too high internal temperature, thanks to the Oxley predictive model. The concerned model provides insights on chip formation under some cutting conditions. A sample of "normal" cutting conditions are first randomly simulated. In order to train the AE model on the resulting set of data, all the input cutting parameters are taken in a range that reflects a realistic machining context: $V \in [50, 300], \alpha \in [-10, 10], w \in [0.5, 3], t_1 \in [0.1, 0.5]$. In order to simulate abnormal cutting conditions, another dataset is also randomly generated, but including out of range parameters: for simplicity, and also for visualization purpose, only two parameters (among the four inputs) are chosen according to this rule. The angle parameters are taken in the range $[-30, 30]$, and the cutting speed in the range $[10, 1000]$, while the two other parameters are chosen in their respective "normal" range (as defined above). A theoretical result of thermo-mechanical laws in metal cutting suggests that a cutting angle taken out of the "normal" range (i.e. $[-10, 10]$) may produce abnormal machining features: for instance, a high angle may generate an internal temperature which is too high, close to melting point. This theoretical assumption is used for defining a rule decision characterizing anomaly introduced in the second dataset.

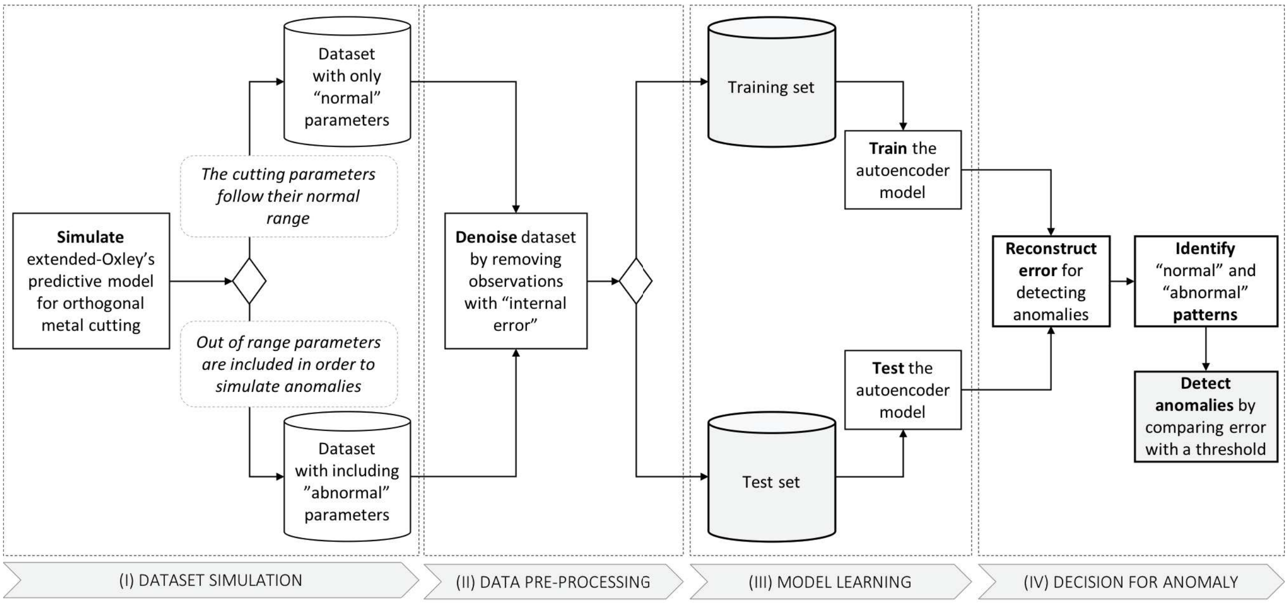

Fig. 4. The different phases of the proposed methodology.

The second phase deals with ***data pre-processing***. As in any machine learning method, before training a model, one need to handle some issues aiming at checking the quality of the data acquired. In some cases, missing values are imputed using appropriate methods; in others, data at hand are cleaned in order to remove noise that is not of interest to the analyst. In the light of the simulation approach for acquiring data adopted, the present study is not concerned with imputation purpose, since all the data are obtained mathematically, through the formulation of relationships among cutting parameters and machining features, thanks to the Oxley predictive model. However, the data simulated (during the first phase of the proposed methodology) needed to be denoised, before their use in learning and testing the AE model. The reason is that, in order for the results of the optimization problem of the Oxley predictive model to be valid, data that do not satisfy an internal error test have to be removed.

The third phase provides the ***learning process*** of the autoencoder model. The dataset with "normal" cutting conditions is used here as the training set, while the one composed of "abnormal" parameters is used for testing the model. The main choices for defining the concerned model are explained further.

The last phase of the proposed methodology is concerned with ***decision-making***, regarding the anomaly detection of the cutting conditions. In that purpose, after training and testing the model (as in the previous phase), the next step is to reconstruct error by comparing the predicted values from the set containing "abnormal" parameters (i.e. the test set) with the pattern of the "normal" data. Then, the corresponding patterns of normal and abnormal errors are visualized in a graph which highlights data that do not follow the "normal" pattern.

B. *Main results and discussion*

The proposed methodology, particularly its associated model (i.e. an autoencoder), has been implemented in Python programming language, which provides useful and powerful packages, including those allowing linear algebra computations. The main results and corresponding discussion are presented in the following.

Phase (I): the training and test sets that have been simulated are (respectively) of size 5000 and 500.

Phase (II): the training and test sets originally generated have then been denoised, in order to keep only valid optimized cutting forces (as explained earlier), using an internal error test in the optimization algorithm of the extended-Oxley analytical model. A summary of the test set is presented in TABLE I, which illustrates the anomalies simulated in this dataset. Indeed, the maximum value of the parameter V is equal to *1000*, while the one associated to α is equal to *29.1*, which are out of their respective "normal" range (as described above).

Phase (III): an autoencoder model has been defined with the following main choices:

- Input and output layers are of size 4 (which is the number of the cutting parameters defined above);
- In order to consider uncertainties due to the machining environment (beside the 4 defined above), 10 nodes have been used in the hidden layer; as explained in the

sub-section A of section III, this allows to implicitly consider latent factors such as surrounding temperature in the plant in the learning model;

TABLE I. SUMMARY OF THE TEST SET

	V(m/min)	*α (degree)*	*w(mm)*	*t_1(mm)*
mean	411.01269	5.779188	1.888325	0.28802
std	275.058066	6.435221	0.753844	0.114311
min	10	-3	0.5	0.1
max	1000	29.1	3	0.5

- *tanh* has been used as activation function for the hidden layer, while linear identity function has been defined for the output;
- Mean squared error (commonly used in regression problems) has been adopted for the loss function, which this is consistent with the present context, since the machining features (such as the cutting force and the internal temperature) are predicted from the cutting parameters (i.e. cutting speed, cutting angle, etc.), using relationships between numerical variables;
- To avoid overfitting, a constant lasso penalty has been added to the loss function *(l_1 = 1e-5)*; a sparse penalty has also been considered in order to help the model to perform well by forgetting some nodes in the hidden layer which has been taken higher than the input *(sparsity = 1e-7)*.

Let us recall that the main focus of this study is not to automate the monitoring of a machining process, but to analyze the feasibility of the proposed method and model for detecting anomalies in the context of orthogonal metal cutting, all the hyper-parameters of the autoencoder have been tuned manually. The reason behind this choice is that the method adopted (an autoencoder) is not a typical one for anomaly detection; so, our main focus was to study its ability to perform such analysis. The parameters presented above are the best found manually (among the various tests realized), and produced a loss function equal to *8.83e-5*. An overview of the quality of the learning process is illustrated in Fig. 5. An improvement of the hyper-parameters tuning, based on hybridization of the proposed model with a combinatorial optimization algorithm, or using a benchmarking process, is planned for further studies.

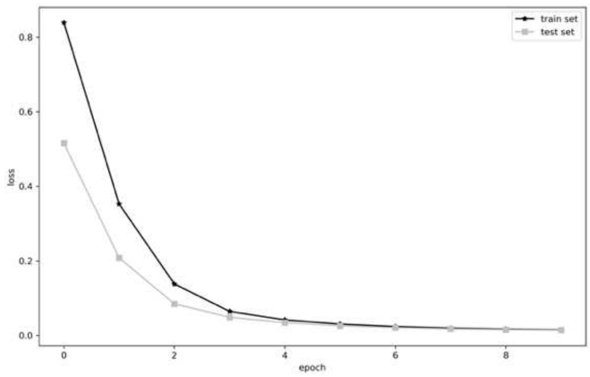

Fig. 5: Learning curves of the training and test sets.

Phase (IV): after training and testing the model, a reconstruction error process has been performed. Samples of initial and reconstructed errors are presented in TABLE II and TABLE III (respectively), which provide an idea on the quality of the reconstruction. The data presented in these two tables are the scaled values of the original parameters (of the training and test sets). Visualization of the anomalies detected is illustrated in Fig. 6, which shows the patterns of the initial values and their reconstructed errors. The parameters declared abnormal are summarized in Table IV, which also presents their corresponding machining features F_c, F_t, T_{int} (cutting force, feed force, internal temperature).

TABLE II. SAMPLE OF INITIAL INPUT CUTTING PARAMETERS

V	α	w	t_1
1.5559900	-0.152161	-1.045740	-1.64482
1.3196800	-1.084530	-0.382473	-0.507566
0.2126360	-0.882516	0.280794	-0.770008
-0.0673047	-1.348700	1.474670	-1.11993
-0.0582157	-0.354174	-1.178390	0.454722
-1.0434600	-1.36424	0.413448	-0.332604
1.0560900	0.547116	0.280794	0.367241
-1.3415800	2.36524	0.148141	0.979607
2.0777000	0.00323411	0.0154874	-1.03245
0.3089800	0.73359	0.944061	0.804645

These results show that abnormal conditions are associated with high cutting angles (see the second column of TABLE IV) and relate to high internal temperatures (see the last column of the TABLE IV), close to the melting point; which is an interesting insight for operators in the plant, in order for them to prevent any failure in the cutting conditions definition.

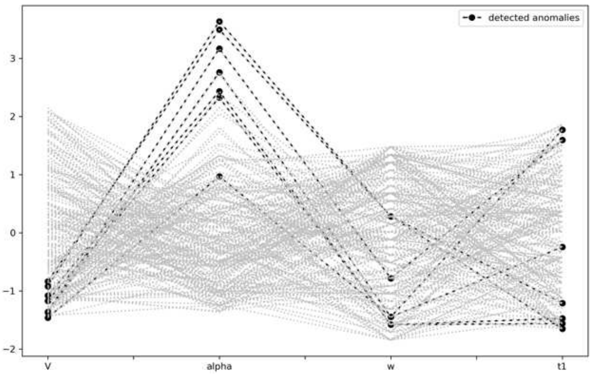

Fig. 6. Cutting conditions pattern revealing abnormal parameters detected by the AE method.

One problem with these first results is that they do not reveal abnormal cutting speed; but, after few more trainings, the model has been able to provide them. The main reason of this lack is that a neural network-based model requires to many hyper-parameters to set; since we have manually tuned them, it is possible that the values used in the model are not optimized, which constitutes an improvement planned for further studies (as mentioned above).

In order to roughly assess the quality of the proposed model and analyze its ability to detect efficiently anomalies in a dataset, a comparison has been realized with one of the most powerful method for outliers' detection used in the literature: Isolation Forest (IF) [25]. While autoencoder, as many neural network-based methods, requires many hyper-parameters to tune for a good result, IF seems less complex to apply: in contrast to other popular methods, IF identify anomalies instead of profiling normal data. With the dataset used in this study, it has been able to identity more anomalies than the first results provided by AE, as illustrated in Fig. 7, which shows that abnormal cutting speed (high values) have been detected. But one drawback of IF forest is that it requires to set a contamination parameter, which represents the percentage of outliers in the dataset; which is not given or known before launching the analysis.

TABLE III. SAMPLE OF RECONSTRUCTED CUTTING PARAMETERS

reconstr_V	**reconstr_α**	**reconstr_w**	**reconstr_t_1**
1.5667400	-0.140437	-1.09505	-1.6198
1.3513800	-1.07248	-0.400129	-0.489684
0.2437020	-0.898479	0.299828	-0.778925
-0.0627466	-1.31954	1.43489	-1.14587
-0.0426192	-0.375626	-1.2255	0.492997
-1.0440900	-1.32003	0.444414	-0.328182
1.1083000	0.577719	0.300205	0.402484
-1.282190 0	2.19836	0.183383	0.948588
1.9724200	0.0179206	0.0116647	-1.02147
0.3370880	0.764873	0.963853	0.830861

TABLE IV. ANOMALIES DETECTED

V	**α**	**w**	**t_1**	**F_c**	**F_t**	**T_{int}**
181.0	**28.2**	2.1	0.10	649.40	545.85	**1174.49**
37.5	**26.1**	1.3	0.47	1982.57	1776.18	**1171.00**

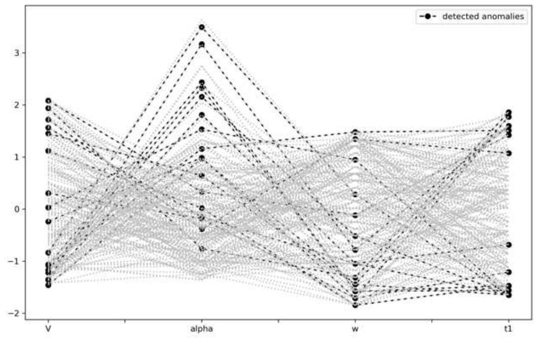

Fig. 7. Cutting conditions pattern revealing abnormal parameters detected by the IF method.

V. CONCLUSION AND PERSPECTIVES

Providing machining companies with decision support allowing them to monitoring their process is becoming crucial in order for them to save maintenance cost and therefore be competitive in a globalized market. In this paper, an autoencoder model for detecting abnormal cutting conditions, in the context of orthogonal metal cutting, has been proposed. A simulation-based approach has been adopted for that purpose, which allowed us to train the model with massive data, in contrast to most of the literature works that are based on few data acquired from measurement devices. In order to reflect a realistic context, the architecture of the proposed autoencoder has been defined with more nodes in the hidden layer than the input and the output. With such a choice, the resulting model implicitly takes into account latent variables such as the environmental temperature of the plant, beside the four cutting parameters considered (cutting speed, cutting angle, cutting width, feed depth). The model implemented provide good results, and has been able to detect the anomalies in the dataset, but after many rounds of pre-training. Compared to Isolation Forest (one of the most powerful method for outlier's detection), the results are equivalent, unless the fact that IF requires less parameter to tune and provides good results in the first run. Two main improvements can be considered and are planned for further studies: (1) provide an operator with knowledge allowing him to directly choose the "optimal" conditions, which may be useful to prevent anomalies, and (2) define a process for optimizing the choice of the hyper-parameters of the autoencoder.

REFERENCES

[1] Oxley, P.L.B. The mechanics of machining: An analytical approach to assessing machinability. (Ellis Horwood, 1989).

[2] Kovač, P., Mankova, I., Gostimirović, M., Sekulić, M., & Savković, B. (2011). A review of machining monitoring systems. Journal of production engineering, 14(1), 1-6.

[3] Oraby, S., & Hayhurst, D. (2004). Cutting tool condition monitoring using surface texture via neural network. International Journal of Mathematical and Computational Tools Manufacturing, 44, 1261–1269.

[4] Teti, R., Jemielniak, K., O'Donnell, G., Dornfeld, D. Advanced monitoring of machining operations, CIRP Annals - Manufacturing Technology 59 (2010) 717–739.

[5] An, J., & Cho, S. (2015). Variational autoencoder based anomaly detection using reconstruction probability. Special Lecture on IE, 2, 1-18.

[6] Douglas M Hawkins. Identification of outliers, volume 11. Springer, 1980.

[7] Byrne G, Dornfeld D, Inasaki I, Konig W, Teti R (1995) Tool Condition Monitoring – The Status of Research and Industrial Application. CIRP Annals 44(2):541–567.

[8] Jantunen, E. (2002). A summary of methods applied to tool condition monitoring in drilling. International Journal of Machine Tools and Manufacture, 42(9), 997-1010.

[9] Varun Chandola, Arindam Banerjee, and Vipin Kumar. Anomaly detection: A survey. ACM computing surveys (CSUR), 41(3):15, 2009.

[10] Hawkins, S., He, H., Williams, G. J., and Baxter, R. A. 2002. Outlier detection using replicator neural networks. In Proceedings of the 4th International Conference on Data Warehousing and Knowledge Discovery. Springer-Verlag, 170–180.

[11] Williams, G., Baxter, R., He, H., Hawkins, S., and Gu, L. 2002. A comparative study of RNN for outlier detection in data mining. In Proceedings of the IEEE International Conference on Data Mining. IEEE Computer Society, 709.

[12] Wong, W.-K., Moore, A., Cooper, G., and Wagner, M. 2003. Bayesian network anomaly pattern detection for disease outbreaks. In Proceedings of the 20th International Conference on Machine Learning. AAAI Press, 808–815.

[13] Sebyala, A. A., Olukemi, T., and Sacks, L. 2002. Active platform security through intrusion detection using naive Bayesian network for anomaly detection. In Proceedings of the London Communications Symposium.

[14] Diehl, C. and Hampshire, J. 2002. Real-time object classification and novelty detection for collaborative video surveillance. In Proceedings of the IEEE International Joint Conference on Neural Networks. IEEE.

[15] Davy, M. and Godsill, S. 2002. Detection of abrupt spectral changes using support vector machines, an application to audio signal segmentation. In Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing.

[16] King, S., King, D., P. Anuzis, K. A., Tarassenko, L., Hayton, P., and Utete, S. 2002. The use of novelty detection techniques for monitoring high-integrity plant. In Proceedings of the International Conference on Control Applications. vol. 1., 221–226.

[17] Byers, S. D. and Raftery, A. E. 1998. Nearest neighbor clutter removal for estimating features in spatial point processes. J. Amer. Statis. Assoc. 93, 577–584.

[18] Sun, P., Chawla, S., and Arunasalam, B. 2006. Mining for outliers in sequential databases. In Proceedings of the SIAM International Conference on Data Mining.

[19] Sun, P. and Chawla, S. 2006. SLOM: A new measure for local spatial outliers. Knowl. Inform. Syst. 9, 4, 412–429.

[20] Ramadas, M., Ostermann, S., and Tjaden, B. C. 2003. Detecting anomalous network traffic with self- organizing maps. In Proceedings of the Conference on Recent Advances in Intrusion Detection. 36–54.

[21] Brockett, P. L., Xia, X., and Derrig, R. A. 1998. Using Kohonen's self-organizing feature map to uncover automobile bodily injury claims fraud. J. Risk Insur. 65, 2, 245–274.

[22] He, Z., Xu, X., and Deng, S. 2003. Discovering cluster-based local outliers. Pattern Recog. Lett. 24, 9–10, 1641–1650.

[23] LeCun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). Gradient-based learning applied to document recognition. Proceedings of the IEEE, 86(11), 2278-2324.

[24] Goodfellow, I., Bengio, Y., & Courville, A. (2016). Deep learning (adaptive computation and machine learning series). Adaptive Computation and Machine Learning series, 800.

[25] Liu, F. T., Ting, K. M., & Zhou, Z. H. (2008, December). Isolation forest. In Data Mining, 2008. ICDM'08. Eighth IEEE International Conference on (pp. 413–422). IEEE